

2.2.3. *Xây dựng hàm ẩn danh với hàm do người dùng định nghĩa (User define function)*

18/. Số Palindrome

Số Palindrome (hoặc Palindromic)

- (Theo Wikipedia) Là một số vẫn giữ nguyên giá trị khi các chữ số của nó được đảo ngược. Hay nói cách khác là số đối xứng.
- 30 số thập phân palindrome đầu tiên là: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 22, 33, 44, 55, 66, 77, 88, 99, 101, 111, 121, 131, 141, 151, 161, 171, 181, 191, 202, ...



Số nguyên tố Palindrome

- (Theo Wikipedia) Là số nguyên tố viết xuôi hay viết ngược vẫn chỉ cho ra một số.
- Các số nguyên tố Palindrome dưới 20000 gồm: 2, 3, 5, 7, 11, 101, 131, 151, 181, 191, 313, 353, 373, 383, 727, 757, 787, 797, 919, 929, 10301, 10501, 10601, 11311, 11411, 12421, 12721, 12821, 13331, 13831, 13931, 14341, 14741, 15451, 15551, 16061, 16361, 16561, 16661, 17471, 17971, 18181, 18481, 19391, 19891, 19991.

Yêu cầu: Sử dụng lambda, viết chương trình

- a. In ra các số Palindrome từ 2 đến 1000000.
- b. In ra các số nguyên tố Palindrome từ 2 đến 100000.

```
import math

def isPalindrome(n):
    '''dx=int(str(n)[::-1])
    if n==dx:
        return True
    else:
        return False'''
    return n==int(str(n)[::-1])

#-----

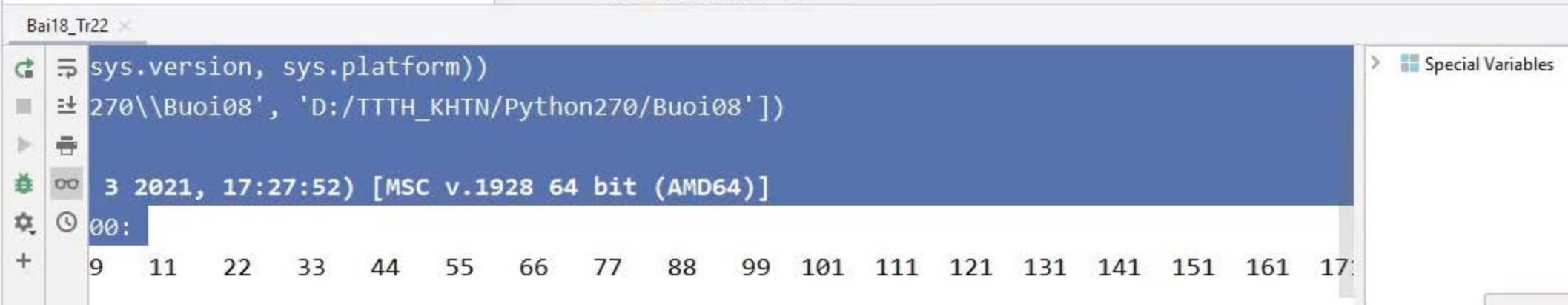
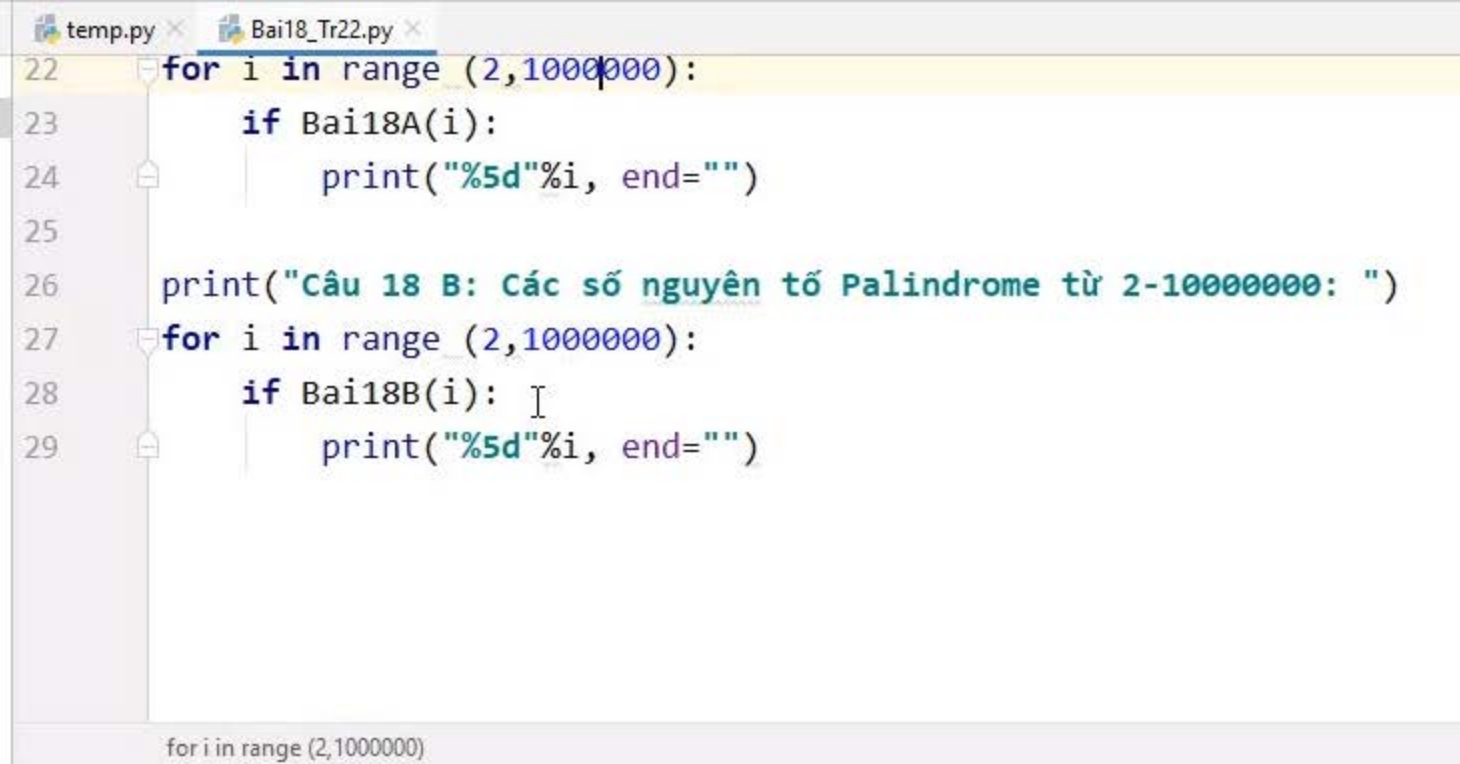
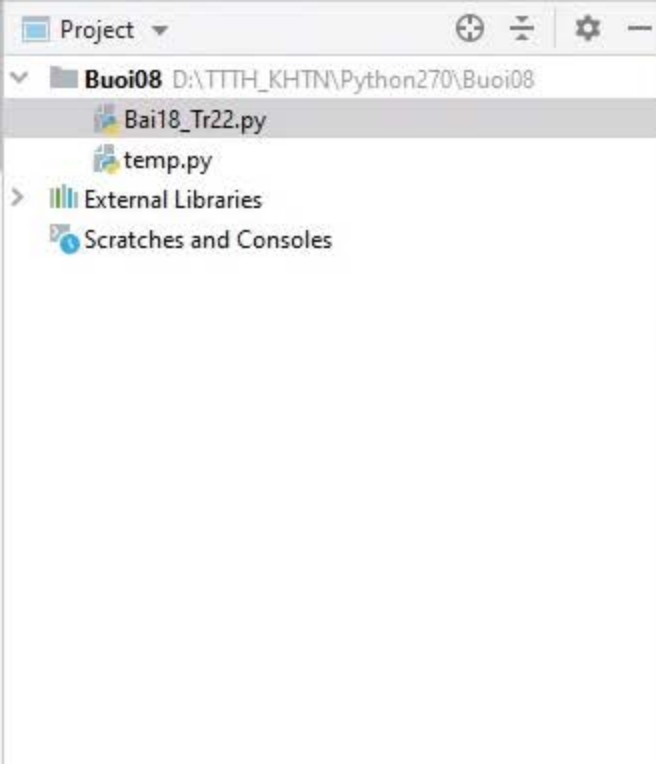
def isPrime(n):
    if n<=1:
        return False
    for i in range (2, int(math.sqrt(n))+1):
        if n%i==0:
            return False
    return True

#-----

Bai18A= lambda x: True if isPalindrome(x) else False
Bai18B= lambda x: True if isPalindrome(x) and isPrime(x) else False

#----- CHUONG TRINH CHINH
```

```
16     return True
17     #-----
18     Bai18A= lambda x: True if isPalindrome(x) else False
19     Bai18B= lambda x: True if isPalindrome(x) and isPrime(x) else False
20     #----- CHUONG TRINH CHINH
21     print("Câu 18 A: Các số Palindrome từ 2-10000000: ")
22     for i in range (2,10000000):
23         if Bai18A(i):
24             print("%5d"%i, end="")
25
26     print("Câu 18 B: Các số nguyên tố Palindrome từ 2-10000000: ")
27     for i in range (2,10000000):
28         if Bai18B(i):
29             print("%5d"%i, end="")
```



37/. Cho 1 list (*lst*) chứa các số nguyên.

- Viết hàm *FindNumber* với tham số truyền cho hàm là số nguyên k . Hàm trả về 3 giá trị nguyên là pre , k , $next$:
 - pre là số chính phương nhỏ hơn và gần với k nhất. Quy ước, nếu $k \leq 1$ thì $pre = 0$.
 - $next$ là số chính phương lớn hơn và gần với k nhất. Quy ước, nếu $k < 1$ thì $next = 1$.
- Trong chương trình chính, sử dụng hàm map để tạo ra 1 list chứa các bộ 3 số do hàm *FindNumber* trả về khi nhận các giá trị của list *lst*.

Ví dụ: List ban đầu $lst = [8, -2, 5, 9]$

Sẽ xuất ra: Kết quả thực hiện: $[(4, 8, 9), (0, -2, 1), (4, 5, 9), (4, 9, 16)]$

```
1  import math
2  def isSquareNumber(k):
3      '''m=int(math.sqrt(k))
4      return m**2==k'''
5      return int(math.sqrt(k)) ** 2 == k
6
7  def FindNumber(k):
8      #B1: tìm pre
9      if k<=1:
10         pre=0
11     else: # k>1
12         #B2: tìm next
```



```
6
7 def FindNumber(k):
8     #B1: tim pre
9     if k<=1:
10         pre=0
11     else: # k>1
12         pre=k-1
13         while isSquareNumber(pre)==False:
14             pre-=1
15     #B2: Tim next
16     next=k+1
17     while isSquareNumber(next)==False:
18         next+=1
19     return pre,k,next
20
```

```
def FindNumber(k):
```

```
    #B1: tim pre
```

```
    if k<=1:
```

```
        pre=0
```

```
    else: # k>1
```

```
        pre=k-1
```

```
        while isSquareNumber(pre)==False:
```

```
            pre-=1
```

```
    #B2: Tim next
```

```
    next=k+1
```

```
    while isSquareNumber(next)==False:
```

```
        next+=1
```

```
    return pre,k,next
```





```
    #-----
```

```
lst=[8,-2,5,9]
```

```
resultList = list(map(FindNumber, lst))
```





```
print(resultList)
```

Buoi08 D:\TTTH_KHTN\Python270\Buoi08

 Bai18_Tr22.py Bai37_Tr34.py temp.py External Libraries Scratches and Consoles

```
1 import math
2 def isSquareNumber(k):
3     '''m=int(math.sqrt(k))
4     return m**2==k'''
5     if k<=0:
6         return False
7     return int(math.sqrt(k)) ** 2 == k
8
9 def FindNumber(k):
    isSquareNumber() > if k<=0
```

Bai18_Tr22 × Bai37_Tr34 ×

  "C:\Program Files\Python39\python.exe" "C:\Program Files\JetBrains\PyCharm Community I import sys; print('Python %s on %s' % (sys.version, sys.platform)) sys.path.extend(['D:\\TTTH_KHTN\\Python270\\Buoi08', 'D:/TTTH_KHTN/Python270/Buoi08']) Python 3.9.5 (tags/v3.9.5:0a7dcdb, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]

[(4, 8, 9), (0, -2, 1), (4, 5, 9), (4, 9, 16)]

In[3]: |

Buoi08

Bai37_Tr34.py

Project

Buoi08

D:\TTTH_KHTN\Python270\Buoi08

Bai18_Tr22.py

Bai37_Tr34.py

temp.py

External Libraries

Scratches and Consoles

temp.py

Bai37_Tr34.py

Bai18_Tr22.py

```
16         pre-=1
17         #B2: Tìm next
18         next=k+1
19         while isSquareNumber(next)==False:
20             next+=1
21         return pre,k,next
22     #-----
23     lst=[8,-2,5,9]
24     print("List ban đầu: ",lst)
25     resultList = list(map(FindNumber, lst))
26     print("List kết quả: ",resultList)
```

Bai18_Tr22

Bai37_Tr34

sys.path.extend(['D:\\TTTH_KHTN\\Python270\\Buoi08', 'D:/TTTH_KHTN/Python270/Buoi08'])

Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]

List ban đầu: [8, -2, 5, 9]

List kết quả: [(4, 8, 9), (0, -2, 1), (4, 5, 9), (4, 9, 16)]

In[3]:

38/. Cho người dùng nhập 1 danh sách tên các màu khác nhau bởi dấu phẩy (,), ví dụ: Red, Blue, Black, White, Pink. Viết chương trình cho biết những ký tự:

- Có xuất hiện trong danh sách (không phân biệt ký tự hoa/thường).
- Không xuất hiện trong danh sách (không phân biệt ký tự hoa/thường).

✍ Ví dụ:

Danh sách ban đầu: ['Red', 'Blue', 'Black', 'White', 'Pink']

Những ký tự xuất hiện trong list ban đầu: a b c d e h i k l n p r t u w

Những ký tự không có trong list ban đầu: f g j m o q s v x y z

✍ Gợi ý: sử dụng hàm map để chuyển mỗi chuỗi trong danh sách thành 1 list, từ đó mới xác định những ký tự có hoặc không có xuất hiện.

Project

temp.py

Bai38_34.py

Bai37_Tr34.py

Bai18_Tr22.py

Buoi08

D:\TTTH_KHTN\Pyt

Bai18_Tr22.py

Bai37_Tr34.py

Bai38_34.py

temp.py

External Libraries

Scratches and Consoles

1

lst=['Red', 'Blue', 'Black','White', 'Pink']

2

print(lst)

3

resultList=[]

4

for item in lst:

5

for kytu in item:

6

if kytu.lower() not in resultList:

7

resultList.append(kytu.lower())

8

resultList.sort()

9

print("Các ký tự có xuất hiện trong list ban đầu:", resultList)

10

for item in lst

>

for kytu in item

>

if kytu.lower() not in resultLi...

Bai38_34

sys.path.extend(['D:\\TTTH_KHTN\\Python270\\Buoi08', 'D:/TTTH_KHTN/Python270/Buoi08'])

Python 3.9.5 (tags/v3.9.5:0a7dcdbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]In[2]: runfile('D:/TTTH_KHTN

['Red', 'Blue', 'Black', 'White', 'Pink']

Các ký tự có xuất hiện trong list ban đầu: ['a', 'b', 'c', 'd', 'e', 'h', 'i', 'k', 'l', 'n', 'p', 'r', 't', 'u',

In[3]:

```
1 lst=['Red', 'Blue', 'Black','White', 'Pink']
2 print(lst)
3 resultList=[]
4 for item in lst:
5     for kytu in item:
6         if kytu.lower() not in resultList:
7             resultList.append(kytu.lower())
8 resultList.sort()
9 print("Các ký tự có xuất hiện trong list ban đầu:", resultList)
10 #-----
11 lowerCase=['a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z']
12 print("Các ký tự không xuất hiện trong list ban đầu:")
13 for kytu in lowerCase:
14     if kytu not in resultList:
15         print(kytu, end=" ")
16
```

1: Project

temp.py

Bai38_34.py

Bai37_Tr34.py

Bai18_Tr22.py

11

lowerCase=['a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z']

12

print("Các ký tự không xuất hiện trong list ban đầu:")

13

for kytu in lowerCase:

14

if kytu not in resultList:

15

print(kytu, end="| ")

16

for kytu in lowerCase → if kytu not in resultList

Bai38_34

Bai38_34 (1)

Python 3.9.5 (tags/v3.9.5:0a7dcdb, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]In[2]: runfile('D:/TT

['Red', 'Blue', 'Black', 'White', 'Pink']

Các ký tự có xuất hiện trong list ban đầu: ['a', 'b', 'c', 'd', 'e', 'h', 'i', 'k', 'l', 'n', 'p', 'r', 't

Các ký tự không xuất hiện trong list ban đầu:

f g j m o q s v x y z

In[3]:

01 item = {str}

01 kytu = {str}

1 lowerCase =

2 lst = {list: 5}

3 resultList =

4 Special Vari

3.7. Xây dựng hàm ẩn danh (*Anonymous Function*) cho Iterator object

3.7.1. Xây dựng hàm ẩn danh trên number list

86/. Cho 2 list chứa các số nguyên và 2 list này có cùng số lượng phần tử. Sử dụng lambda để tạo ra 1 list mới bằng cách cộng đôi một các số có trong 2 list.

👉 Gợi ý: sử dụng hàm map

Ví dụ: `lst1 = [1, 2, 3]`, `lst2 = [4, 5, 6]` => in ra màn hình `[5, 7, 9]`

lst1 =	[1, 2, 3]
lst2 =	[4, 5, 6]
resultList =	[5, 7, 9]

```
1 lst1=[1,2,3,9]
2 lst2=[4,5,6]
3 print(lst1)
4 print(lst2)
5 lst3= list(map(lambda x,y: x+y,lst1,lst2))
6 print(lst3)
7
```


87/. Cho 3 list chứa các số nguyên có cùng số lượng phần tử. Tạo ra 1 list mới bằng cách chọn số lớn nhất trong bộ các phần tử có cùng thứ tự (index) có trong 3 list.

lst1 =	[1	,	2	,	3	,	9]
lst2 =	[4	,	8	,	6	,	12]
lst3 =	[7	,	5	,	10	,	11]
<hr/>									
resultList =	[7	,	8	,	10	,	12]

Ví dụ: với 3 list cho trước: $lst1 = [1, 2, 3, 9]$, $lst2 = [4, 8, 6, 12]$, $lst3 = [7, 5, 10, 11] \Rightarrow$ in ra màn hình $[7, 8, 10, 12]$.

Project

Buoi08

D:\TTTH_KHTN\Pyt

Bai18_Tr22.py

Bai37_Tr34.py

Bai38_34.py

Bai86_Tr46.py

Bai87_Tr46.py

temp.py

External Libraries

Scratches and Consoles

temp.py

Bai38_34.py

Bai86_Tr46.py

Bai87_Tr46.py

Bai37_Tr34.py

Bai18_Tr22.py

3

4

5

6

7

8

9

10

lst3=[7,5,10,11]

print(lst1)

print(lst2)

print(lst3)

lst4= list(map(lambda x,y,z: max(x,y,z), lst1, lst2, lst3))

print(lst4)

Bai38_34

Bai38_34 (1)

Bai86_Tr46

Bai87_Tr46

Python 3.9.5 (tags/v3.9.5:0a7dcdb, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]

[1, 2, 3, 9]

[4, 8, 6, 12]

[7, 5, 10, 11]

[7, 8, 10, 12]

In[3]:

88/. Cho list1 chứa các số nguyên bất kỳ. Sử dụng lambda để:

- a. Từ list1, tạo list2 chứa các số chẵn, list3 chứa các số lẻ. In 2 list này ra màn hình.

Ví dụ list1 = [-5, 10, -3, -1, 7, 8, 9, 2]

In ra màn hình: List ban đầu: [-5, 10, -3, -1, 7, 8, 9, 2]

Các số chẵn có trong list: [10, 8, 2]

Các số lẻ có trong list: [-5, -3, -1, 7, 9]

- b. Tương tự câu a nhưng lần lượt các lambda chỉ tính số lượng số chẵn, số lượng số lẻ. In kết quả thực hiện ra màn hình.

👉 Gợi ý: sử dụng hàm filter khi tạo 2 list mới

- c. Tạo list2 chứa các giá trị là bình phương của các giá trị có trong list1 và list3 chứa các giá trị là lũy thừa 3 của các giá trị có trong list1. In kết quả ra màn hình.

👉 Gợi ý sử dụng hàm map khi tạo 2 list mới

Ví dụ

List ban đầu: [-5, 10, -3, -1, 7, 8, 9, 2]

Bình phương các giá trị trong lst: [25, 100, 9, 1, 49, 64, 81, 4]

Lũy thừa 3 các giá trị trong lst: [-125, 1000, -27, -1, 343, 512, 729, 8]

Project | temp.py | Bai38_34.py | Bai86_Tr46.py | Bai87_Tr46.py | Bai37_Tr34.py | Bai18_Tr22.py

Buoi08 D:\TTTH_KHTN\Pyt

- Bai18_Tr22.py
- Bai37_Tr34.py
- Bai38_34.py
- Bai86_Tr46.py
- Bai87_Tr46.py
- temp.py

External Libraries

Scratches and Consoles

```
1 # c88trg46
2 lst1 = [-5,10,-3,-1,7,8,9,2]
3 lstChan = list(filter(lambda x : x%2==0, lst1))
4 print("List chan: ",lstChan)
5 lstLe = list(filter(lambda x : x%2!=0, lst1))
6 print("List le: ",lstLe)
7
```

Bai38_34 | Bai38_34 (1) | Bai86_Tr46 | Bai87_Tr46 | temp

```
"C:\Program Files\Python39\python.exe" "C:\Program Files\JetBrains\PyCharm Community I
import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['D:\TTTH_KHTN\Python270\Buoi08', 'D:/TTTH_KHTN/Python270/Buoi08'])

Python 3.9.5 (tags/v3.9.5:0a7dcdb, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]
List chan: [10, 8, 2]
List le: [-5, -3, -1, 7, 9]

In[3]:
```


Project

Buoi08

D:\TTTH_KHTN\Pyt

Bai18_Tr22.py

Bai37_Tr34.py

Bai38_34.py

Bai86_Tr46.py

Bai87_Tr46.py

Bai88_Tr46.py

temp.py

External Libraries

Scratches and Consoles

temp.py

Bai88_Tr46.py

Bai38_34.py

Bai86_Tr46.py

Bai87_Tr46.py

Bai37_Tr34.py

Bai18_Tr22.py

```
1 print("Câu A:")
2 lst1 = [-5,10,-3,-1,7,8,9,2]
3 lstChan = list(filter(lambda x : x%2==0, lst1))
4 print("List chan: ",lstChan)
5 lstLe = list(filter(lambda x : x%2!=0, lst1))
6 print("List le: ",lstLe)
7 print("Câu B:")
```

Bai38_34

Bai38_34 (1)

Bai86_Tr46

Bai87_Tr46

temp

Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]

List chan: [10, 8, 2]

List le: [-5, -3, -1, 7, 9]

In[3]:

lst1 = {

lstChan

lstLe =

Special

Buoi08

Bai88_Tr46.py

Project

Buoi08

D:\TTTT_KHTN\Pyt

Bai18_Tr22.py

Bai37_Tr34.py

Bai38_34.py

Bai86_Tr46.py

Bai87_Tr46.py

Bai88_Tr46.py

temp.py

External Libraries

Scratches and Consoles

temp.py

Bai88_Tr46.py

Bai38_34.py

Bai86_Tr46.py

Bai87_Tr46.py

Bai37_Tr34.py

Bai18_Tr22.py

1

2

3

4

5

6

7

8

9

10

11

12

13

14

```
print("Câu A:")  
lst1 = [-5,10,-3,-1,7,8,9,2]  
lstChan = list(filter(lambda x : x%2==0, lst1))  
print("List chan: ",lstChan)  
lstLe = list(filter(lambda x : x%2!=0, lst1))  
print("List le: ",lstLe)  
  
print("Câu B:")  
lst1 = [-5,10,-3,-1,7,8,9,2]  
lstChan = list(filter(lambda x : x % 2==0, lst1))  
print("Luong so chan: ",len(lstChan))  
lstLe = list(filter(lambda x : x % 2!=0, lst1))  
print("Luong so le: ",len(lstLe))
```

Bai38_34

Bai38_34 (1)

Bai86_Tr46

Bai87_Tr46

temp

In[3]:

> lst1 =

> lstChan =

> lstLe =

1: Project

Project

Buoi08

D:\TTH_KHTN\Pyt

Bai18_Tr22.py

Bai37_Tr34.py

Bai38_34.py

Bai86_Tr46.py

Bai87_Tr46.py

Bai88_Tr46.py

temp.py

External Libraries

Scratches and Consoles

temp.py

Bai88_Tr46.py

Bai38_34.py

Bai86_Tr46.py

Bai87_Tr46.py

Bai37_Tr46.py

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

```
print("List chan: ",lstChan)

lstLe = list(filter(lambda x : x%2!=0, lst1))
print("List le: ",lstLe)


print("Câu B:")
lst1 = [-5,10,-3,-1,7,8,9,2]
lstChan = list(filter(lambda x : x % 2==0, lst1))
print("Luong so chan: ",len(lstChan))
lstLe = list(filter(lambda x : x % 2!=0, lst1))
print("Luong so le: ",len(lstLe))


print("Câu C:")
lst2 = list(map(lambda x: x**2,lst1))
print(lst2)
lst3 = list(map(lambda x: x**3,lst1))
print(lst3)
```

lambda (x)

Bai38_34

Bai38_34 (1)

Bai86_Tr46

Bai87_Tr46

temp

In[3]:

>>

>>

```
20
21 # Cau D
```

```
22 import math
```

```
23 def KtraSNT(n):
```

```
24     if n<=1:
```

```
25         return False
```

```
26     for i in range (2, int(math.sqrt(n))+1):
```

```
27         if n%i==0:
```

```
28             return False
```

```
29     return True
```

```
30 # -----
```

```
31 def SchinhPhuong(n):
```

```
32     if n<1:
```

```
33         return False
```

```
34     return (int(math.sqrt(n)) ** 2 == n)
```

```
35
36 lst1 = [19,65,81,39,152,639,121,44,100,31]
```

```
37 lst2 = list(filter(lambda x : SchinhPhuong(x) and KtraSNT(x),lst1))
```

```
38 print(lst2)
39
```


The image shows a Python IDE with a project explorer on the left, a code editor in the center, and a console/output pane at the bottom.

Project Explorer (Left): Shows a project named 'Buoi08' with several Python files: Bai18_Tr22.py, Bai37_Tr34.py, Bai38_34.py, Bai86_Tr46.py, Bai87_Tr46.py, Bai88_Tr46.py, Bai91_Tr47.py, and temp.py. There are also 'External Libraries' and 'Scratches and Consoles' sections.

Code Editor (Center): Displays the content of 'temp.py'. The code is as follows:

```
1 # Câu a
2 lst1 = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
3 print(lst1)
4 lst2 = list(filter(lambda x: len(x)==6, lst1))
5 print('Câu A: ', lst2)
6
7 lst1 = ['php', 'www', 'Python', 'abba', 'Java', 'MADAM']
8 print(lst1)
9 lst2 = list(filter(lambda x: x==x[::-1], lst1))
10 print('Câu B: ', lst2)
```

Console/Output (Bottom): Shows the execution results. The first two lines are 'Câu A: ['Monday', 'Friday', 'Sunday']' and '['php', 'www', 'Python', 'abba', 'Java', 'MADAM']'. The third line is 'Câu B: ['php', 'www', 'abba', 'MADAM']'. Below this, it shows 'In[3]:'.

Taskbar (Bottom): Shows several open tabs: Bai38_34 (1), Bai86_Tr46, Bai87_Tr46, temp, temp (1), Bai91_Tr47, and Bai91_Tr47 (1).

4.8. Sử dụng kỹ thuật Comprehension cho sequence data type

- *Comprehension* là một biểu thức đi kèm với lệnh *for* được đặt trong cặp dấu ngoặc vuông ([]) – khi cần tạo *list*) hoặc cặp dấu ngoặc nhọn ({ }) – khi cần tạo *set* hoặc *dictionary*).
- Nhược điểm của *comprehensions*: tất cả các phần tử sẽ được sinh ra và lưu vào bộ nhớ. Vì vậy, chỉ nên sử dụng *comprehensions* đối với những đối tượng có số lượng phần tử không quá lớn.
- **Cú pháp** sử dụng *Comprehension*:

- Đối với *list*:

```
[f(x) for x in iterable [if condition] ]
```

Hoặc

```
[f(x) for x in iterable1 [f(y) for y in iterable2] [if condition]]
```

- Đối với *set*:

```
{f(x) for x in iterable [if condition] }
```

Hoặc

```
{f(x) for x in iterable1 [f(y) for y in iterable2] [if condition]}
```

3.8. Sử dụng kỹ thuật Comprehension cho cho Iterator object

3.8.1. *Number list comprehensions*

97/. Viết chương trình cho nhập số nguyên n. Sử dụng comprehension để tạo ra 1 list chứa các ước số của n.

Ví dụ: với $n=15$, chương trình sẽ in ra: Các ước số của 15 là: `[1, 3, 5, 15]`

98/. Sử dụng comprehension để thực hiện các yêu cầu sau

a. Cho người dùng nhập lần lượt 8 số nguyên.

Yêu cầu: code sử dụng cho người dùng nhập các số, sử dụng lệnh *input* kèm với *for* theo

dạng: `myList = [int(input("...")) for i in range(...)]`

b. In ra các số nguyên tố có trong 8 số vừa nhập.

```
1 '''lst=[]
2     for i in range(1,4):
3         n=int(input("Nhập số nguyên dương thứ %d: "%i))
4         lst.append(n)
5     '''
6     lst=[ int(input("Nhập số nguyên dương thứ "+str(i)+":")) for i in range(1,4) ]
7     print(lst)
```

"C:\Program Files\Python39\python.exe" "C:\Program Files\JetBrains\PyCharm Community I" > Special Variables

```
import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['D:\\TTTH_KHTN\\Python270\\Buoi08', 'D:/TTTH_KHTN/Python270/Buoi08'])

Python 3.9.5 (tags/v3.9.5:0a7dcdb, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]
Nhập số nguyên dương thứ 1:>? 4
Nhập số nguyên dương thứ 2:
>? |
```



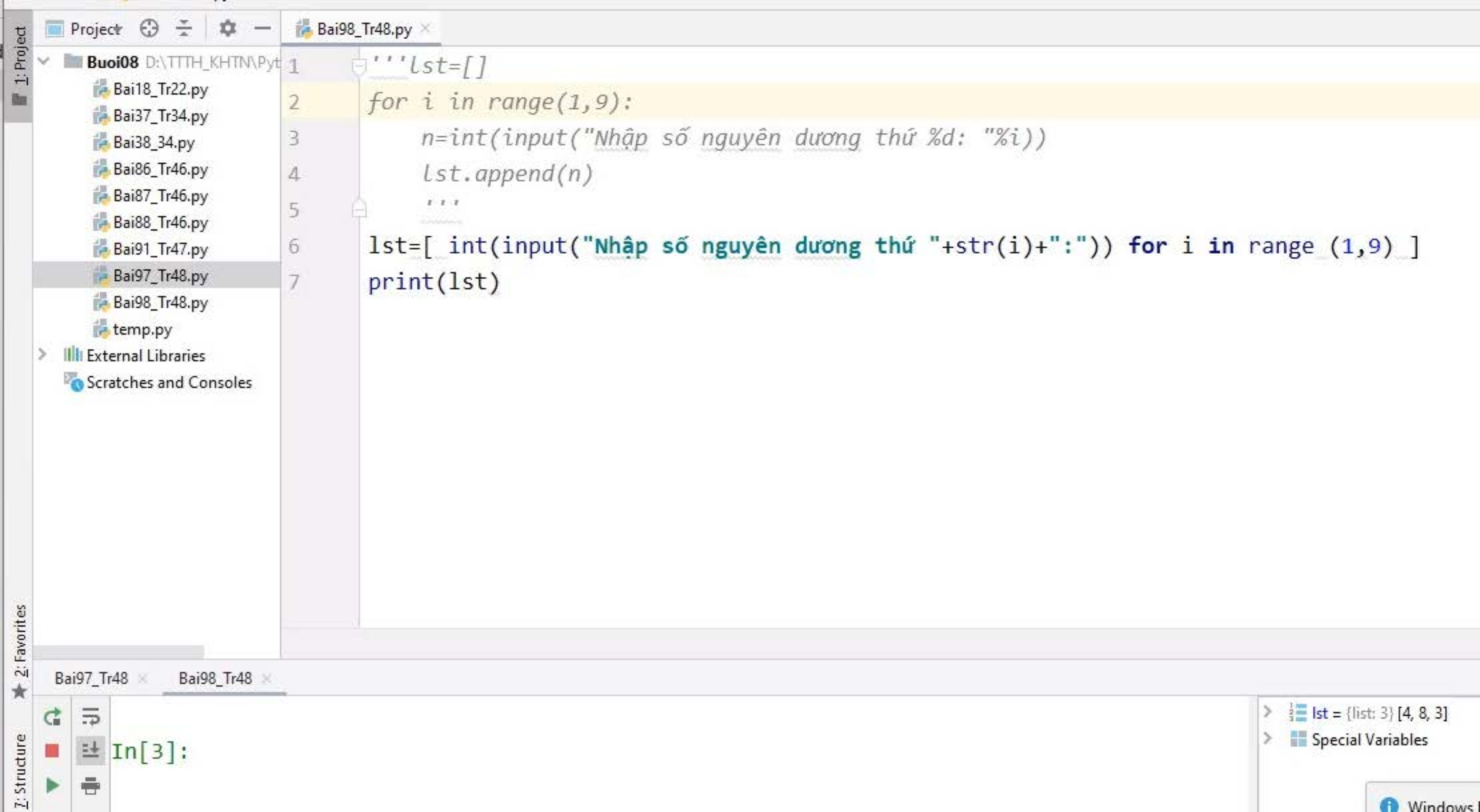
```
1 '''lst=[]
2   for i in range(1,4):
3       n=int(input("Nhập số nguyên dương thứ %d: "%i))
4       lst.append(n)
5   '''
```

*thay thế hết cho các dòng
trên đây*



```
6 lst=[int(input("Nhập số nguyên dương thứ "+str(i)+":")) for i in range (1,4) ]
7 print(lst)  I
```

chỉ cần 1 dòng code là đủ



```
1 def isPrime(n):
2     if n<=1:
3         return False
4     for i in range (2, int(math.sqrt(n))+1):
5         if n%i==0:
6             return False
7     return True
8     #-----
9     '''lst=[]
10    for i in range(1,9):
11        n=int(input("Nhập số nguyên dương thứ %d: "%i))
12        lst.append(n)
13    '''
14    lst=[ int(input("Nhập số nguyên dương thứ "+str(i)+":")) for i in range (1,9) ]
15    print(lst)
16
```

```

8         return True
9         #-----
10        '''lst=[]
11        for i in range(1,9):
12            n=int(input("Nhập số nguyên dương thứ %d: "%i))
13            lst.append(n)
14        '''
15        lst=[ int(input("Nhập số nguyên dương thứ "+str(i)+":")) for i in range (1,9) ]
16        print(lst)
17        lst2=[x for x in lst if isPrime(x)]
18        print("Các số nguyên tố có trong list:",lst2)
19

```

101/. Cho nhập 1 số nguyên n , tạo list L gồm n phần tử với giá trị ngẫu nhiên nguyên dương từ 0 đến 100000.

- a. Cho nhập số k . Sử dụng kỹ thuật Comprehension, cho biết k có lớn hơn tất cả các số có trong L hay không? Nếu không, chương trình xét tiếp xem k có lớn hơn bất kỳ số nào có trong list hay không? Nếu vẫn không có, chương trình in ra '*k nhỏ hơn tất cả các số có trong list*'.

☞ Gợi ý: dùng hàm $all()$ và $any()$.

- b. **Số giảm dần**: Giả sử n được gọi là số giảm dần khi giá trị các ký số có trong n giảm dần **từ trái sang phải**

Ví dụ: số giảm dần như: $n = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 21, 31, 32, \dots, 420, \dots, 77431, \dots$

số KHÔNG giảm dần như: $12, 13, \dots, 557, \dots, 77434, \dots$

☞ Yêu cầu cài đặt: Viết hàm liệt kê các số giảm dần có trong list.

☞ Kết quả gợi ý khi thực hiện chương trình:

Giả sử với list gồm $[727, 6, 1421, 626, 3706, 101, 3553, 4234, 33, 971]$

Hàm sẽ in ra kết quả: Trong list có 3 số giảm dần: 6, 33, 971

Với list gồm $[1599, 1910, 14, 1050, 258, 4387]$

Hàm sẽ in ra kết quả: Trong list không chứa số giảm dần

4.3.7.2. Một số hàm hỗ trợ việc xử lý tuple (giống với list)

- `all()` : Trả về giá trị *True* nếu tất cả các phần tử của *tuple* là *true* hoặc *tuple* rỗng.
- `any()` : Trả về *True* nếu bất kỳ phần tử nào của *tuple* là *True*, nếu *tuple* rỗng trả về *False*.
- `enumerated()`: Trả về đối tượng *enumerate* (liệt kê), chứa cặp *index* và giá trị của tất cả phần tử của *tuple*.
- `len()` : Trả về độ dài (số lượng phần tử) của *tuple*.
- `max()` : Trả về phần tử lớn nhất của *tuple*.
- `min()` : Trả về phần tử nhỏ nhất của *tuple*.
- `sorted()` : Lấy phần tử trong *tuple* và trả về *list* mới được sắp xếp (*tuple* không sắp xếp được).
- `sum()` : Trả về tổng tất cả các phần tử trong *tuple*.

4.6.3. `any()`

- Trả về *True* khi bất kỳ phần tử nào của *iterable* là *True*
- Ví dụ 4.85: xét xem chuỗi `s = 'Sai Gon'` có chứa ít nhất 1 ký tự in thường hay không?

Mã lệnh	Kết quả
<pre>s = 'Sai Gon' print(any(c.islower() for c in s))</pre>	True

4.6.2. *all()*

- Trả về *True* khi tất cả các phần tử trong *iterable* đều có kết quả là *True*.
- Ví dụ 4.83: xét xem chuỗi `s = 'Sai Gon'` có chứa toàn ký tự in thường hay không?

Mã lệnh	Kết quả
<code>s = 'Sai Gon'</code> <code>print(all(c.islower() for c in s))</code>	False
<code>s = 'sai gon'</code> <code>print(all(c.islower() for c in s))</code>	False <i>'''vì ký tự khoảng trắng ở giữa không có trong bộ alphabet'''</i>
<code>s = 'saigon'</code> <code>print(all(c.islower() for c in s))</code>	True

```

1  import math
2  def isPrime(n):
3      if n<=1:
4          return False
5      for i in range (2, int(math.sqrt(n))+1):
6          if n%i==0:
7              return False
8      return True
9  n=1000000
10 cách 1: dem=0; 1-7: chẵn tăng đếm (1,7)=2 số 25 chữ số 3000 nam
11 cách 2: 500001-999999=> 8 không chia chan cho 5-7
12
13 #-----
14 '''lst=[]
15 for i in range(1,9):
16     n=int(input("Nhập số nguyên dương thứ %d: "%i))

```

isPrime() > for i in range (2, int(math.sqr...

bài giảng số nguyên tố

1: Project

Buoi08

D:\TTTH_KHTN\Pyt

Bai18_Tr22.py

Bai37_Tr34.py

Bai38_34.py

Bai86_Tr46.py

Bai87_Tr46.py

Bai88_Tr46.py

Bai91_Tr47.py

Bai97_Tr48.py

Bai98_Tr48.py

temp.py

External Libraries

Scratches and Consoles

Bai98_Tr48.py

Bai18_Tr22.py

```
1 import math
2 def isPrime(n):
3     if n<=1:
4         return False
5     for i in range (2, int(math.sqrt(n))+1):
6         if n%i==0:
7             return False
8     return True
9 n=1000000
10 cách 1: dem=0; 1-7: chẵn tăng đếm (1,7)=2 số 25 chữ số 3000 nam
11 cách 2: 500001-999999=> 8 không chia chan cho 5-7
12 cách 3: 9
13 #-----
14 '''lst=[]
15 for i in range(1,9):
16     n=int(input("Nhập số nguyên dương thứ %d: "%i))

isPrime() > for i in range (2, int(math.sqr...
```

2: Favorites

Bai97_Tr48

Bai98_Tr48

In[3]:


```
1 # c101trg49
2 import random
3 n = int(input("Vui long nhap so n : "))
4 lstL = [random.randint(0,20) for n in range(n)]
5 print(lstL)
6 k = int(input("Vui long nhap so k : "))
7 lst2 = all(k > i for i in lstL)
8 lst3 = any(k > i for i in lstL)
9 if (lst2):
10     print("%d lon hon tat ca cac so"%k)
11 elif (lst3):
12     print("co so nho hon %d trong list" % k)
13 else:
14     print("ko co so nao trong list nho hon %d" % k)
15
```

```
1 # c101trg49
2 import random
3 n = int(input("Vui long nhap so n : "))
4 lstL = [random.randint(0,20) for n in range(n)]
5 print(lstL)
6 k = int(input("Vui long nhap so k : "))
7 kq1 = all(k > i for i in lstL)
8 kq2 = any(k > i for i in lstL)
9 if (kq1):
10     print("%d lon hon tat ca cac so"%k)
11 elif (kq2):
12     print("co so nho hon %d trong list" % k)
13 else:
14     print("ko co so nao trong list nho hon %d" % k)
15
```

b. *Số giảm dần*: Giả sử n được gọi là số giảm dần khi giá trị các ký số có trong n giảm dần **từ trái sang phải**

Ví dụ: số giảm dần như: $n = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 21, 31, 32, \dots, 420, \dots, 77431, \dots$
số KHÔNG giảm dần như: $12, 13, \dots, 557, \dots, 77434, \dots$

➤ *Yêu cầu cài đặt*: Viết hàm liệt kê các số giảm dần có trong list.

➤ *Kết quả gợi ý khi thực hiện chương trình*:

Giả sử với list gồm $[727, 6, 1421, 626, 3706, 101, 3553, 4234, 33, 971]$

Hàm sẽ in ra kết quả: Trong list có 3 số giảm dần: 6, 33, 971

Với list gồm $[1599, 1910, 14, 1050, 258, 4387]$

Hàm sẽ in ra kết quả: Trong list không chứa số giảm dần

Project

Bai101_Tr49.py

temp.py

Buoi08

D:\TTTH_KHTN\Pyt

Bai18_Tr22.py

Bai37_Tr34.py

Bai38_34.py

Bai86_Tr46.py

Bai87_Tr46.py

Bai88_Tr46.py

Bai91_Tr47.py

Bai97_Tr48.py

Bai98_Tr48.py

Bai101_Tr49.py

temp.py

External Libraries

Scratches and Consoles

1

import random

2

n = int(input("Vui long nhap so n : "))

3

lst = [random.randint(0,100000) for n in range(n)]

4

print(lst)

5

lstKQ=[x for x in lst if all(str(x)[k]>=str(x)[k+1] for k in range (len(str(x))-1))]

6

if len(lstKQ)!=0:

7

print("Số giảm dần")

8

else:

9

print("Số KHÔNG giảm dần")

10

11

Bai97_Tr48

Bai98_Tr48

temp

sys.path.extend(['D:\\TTTH_KHTN\\Python270\\Buoi08', 'D:/TTTH_KHTN/Python270/Buoi08'])

Python 3.9.5 (tags/v3.9.5:0a7dcbbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]

Vui long nhap so n : >? 20

[50927, 38924, 77215, 35177, 91530, 40642, 66247, 7739, 9127, 5421, 74422, 64907, 88258, 3263, 50673, 78528, 91530]

Số giảm dần

In[3]: |

Project | temp.py | Bai101_Tr49.py | temp.py

Buoi08 D:\TTTH_KHTN\Pyt

- Bai18_Tr22.py
- Bai37_Tr34.py
- Bai38_34.py
- Bai86_Tr46.py
- Bai87_Tr46.py
- Bai88_Tr46.py
- Bai91_Tr47.py
- Bai97_Tr48.py
- Bai98_Tr48.py
- Bai101_Tr49.py
- temp.py

External Libraries

Scratches and Consoles

```
1 import random
2 n = int(input("Vui long nhap so n : "))
3 lst = [random.randint(0,1000) for n in range(n)]
4 print(lst)
5 lstKQ=[ x for x in lst if all(str(x)[k]>=str(x)[k+1] for k in range (len(str(x))-1))]
6 print(lstKQ)
7
8
9
10
11
```

*trong trường hợp này là
đang so sánh trực tiếp giá trị
dựa trên bảng mã ASCII*

Bai97_Tr48 | Bai98_Tr48 | temp

```
sys.path.extend(['D:\\TTTH_KHTN\\Python270\\Buoi08', 'D:/TTTH_KHTN/Python270/Buoi08'])

Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]

Vui long nhap so n : >? 100

[494, 546, 373, 836, 554, 627, 879, 637, 8, 65, 324, 954, 949, 388, 678, 720, 87, 497, 907, 12, 241, 18, 28, 415, 1
[554, 8, 65, 954, 720, 87, 100, 98, 870, 611, 8, 5, 961, 852, 531, 852, 952, 31, 987, 54, 744, 3, 761]

In[3]:
```

Windows Defender configuration updated


```
Project  Bai101_Tr49.py  temp.py
Buoi08 D:\TTTH_KHTN\Pyt
  Bai18_Tr22.py
  Bai37_Tr34.py
  Bai38_34.py
  Bai86_Tr46.py
  Bai87_Tr46.py
  Bai88_Tr46.py
  Bai91_Tr47.py
  Bai97_Tr48.py
  Bai98_Tr48.py
  Bai101_Tr49.py
  temp.py
> External Libraries
Scratches and Consoles

3 lst = [random.randint(0,1000) for n in range(n)]
4 print(lst)
5 lstKQ=[x for x in lst if all(str(x)[k]>=str(x)[k+1] for k in range (len(str(x))-1))]
6 print(lstKQ)
7
8 range => 0,1,2
9
10 5 5 4
11 0 1 2
12
13
```

```
Bai97_Tr48  Bai98_Tr48  temp
sys.path.extend(['D:\\TTTH_KHTN\\Python270\\Buoi08', 'D:/TTTH_KHTN/Python270/Buoi08'])

Python 3.9.5 (tags/v3.9.5:0a7dcdb, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]

> Vui long nhap so n : >? 100
[494, 546, 373, 836, 554, 627, 879, 637, 8, 65, 324, 954, 949, 388, 678, 720, 87, 497, 907, 12, 241, 18, 28, 415, 1
[554, 8, 65, 954, 720, 87, 100, 98, 870, 611, 8, 5, 961, 852, 531, 852, 952, 31, 987, 54, 744, 3, 761]

In[3]:
```

```
1 import random
2 n = int(input("Vui long nhap so n : "))
3 lst = [random.randint(0,1000) for n in range(n)]
4 print(lst)
5 lstKQ=[ x for x in lst if all(str(x)[k]>=str(x)[k+1] for k in range (len(str(x))-1))]
6 print(lstKQ)
```



```
8 '0' => 48
9 '1' => 49
10 '2' => 50
```

```
7 kq1 = all(k > i for i in lstL)
8 kq2 = any(k > i for i in lstL)
9 if (kq1):
10     print("%d lon hon tat ca cac so"%k)
11 elif (kq2):
12     print("co so nho hon %d trong list" % k)
13 else:
14     print("ko co so nao trong list nho hon %d" % k)
15
16 #câu B
17 import random
18 n = int(input("Vui long nhap so n : "))
19 lst = [random.randint(0,1000) for n in range(n)]
20 print(lst)
21 lstKQ=[ x for x in lst if all(str(x)[k]>=str(x)[k+1] for k in range (len(str(x))-1))]
22 print(lstKQ)
23
```

102/. Tạo 2 listA, listB với kích thước khác nhau và chứa giá trị tùy ý. Ví dụ:

```
x = [1, 2, 3]
y = [5, 10, 15, 20]
```

Viết chương trình tạo ra *listC* bằng cách tính tích của từng phần tử trong x với từng phần tử trong y sao cho listC chỉ chứa những phần tử tích này nhưng loại đi những số chẵn. Ví dụ: khi lần lượt tính tích của từng phần tử trong x và từng phần tử trong y, ta được: [5, 10, 15, 10, 20, 30, 15, 30, 45]. Nhưng kết quả yêu cầu loại bỏ những phần tử chẵn nên listC chỉ còn: [5, 15, 15, 45]

Buoi08 D:\TTTH_KHTN\Pyt

- Bai18_Tr22.py
- Bai37_Tr34.py
- Bai38_34.py
- Bai86_Tr46.py
- Bai87_Tr46.py
- Bai88_Tr46.py
- Bai91_Tr47.py
- Bai97_Tr48.py
- Bai98_Tr48.py
- Bai101_Tr49.py
- Bai102_Tr49.py
- temp.py

External Libraries

i97_Tr48 Bai98_Tr48 temp Bai102_Tr49

```
1 lst1=[1,2,3]
2 lst2=[5,10,15,20]
3 lst3=[x*y for x in lst1 for y in lst2]
4 print(lst3)
```

```
"C:\Program Files\Python39\python.exe" "C:\Program Files\JetBrains\PyCharm Community I
```



```
import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['D:\\TTTH_KHTN\\Python270\\Buoi08', 'D:/TTTH_KHTN/Python270/Buoi08'])
```

```
Python 3.9.5 (tags/v3.9.5:0a7dcdb, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]
```

```
[5, 10, 15, 20, 10, 20, 30, 40, 15, 30, 45, 60]
```

```
In[3]:
```


Project     Bai101_Tr49.py

Bai18_Tr22.py
Bai37_Tr34.py
Bai38_34.py
Bai86_Tr46.py
Bai87_Tr46.py
Bai88_Tr46.py
Bai91_Tr47.py
Bai97_Tr48.py
Bai98_Tr48.py
Bai101_Tr49.py
Bai102_Tr49.py
temp.py

External Libraries

Bai101_Tr49.py Bai102_Tr49.py temp.py

```
1 lst1=[1,2,3]
2 lst2=[5,10,15,20]
3 #lst3=[x*y for x in lst1 for y in lst2 if (x*y)%2!=0 ]
4 lst3=[x*y for x in lst1 if x%2!=0 for y in lst2 if y%2!=0 ]
5 print(lst3)
```

lấy x y lẻ nhân nhau, ko lấy chẵn

I

Project [Icons] [Settings] [Close]

Bai101_Tr49.py x Bai102_Tr49.py x temp.py x

▼ Buoi08 D:\TTTH_KHTN\Pyt

- Bai18_Tr22.py
- Bai37_Tr34.py
- Bai38_34.py
- Bai86_Tr46.py
- Bai87_Tr46.py
- Bai88_Tr46.py
- Bai91_Tr47.py
- Bai97_Tr48.py
- Bai98_Tr48.py
- Bai101_Tr49.py
- Bai102_Tr49.py
- temp.py

> External Libraries

```
1 lst1=[1,2,3]
2 lst2=[5,10,15,20]
3 lst3=[x*y for x in lst1 for y in lst2 if (x*y)%2!=0 ]
4 print(lst3)
```

I
chỉ lấy kết quả lẻ

3.9. Giải quyết 1 yêu cầu bằng cả hai kỹ thuật Comprehension & Anonymous Function

Yêu cầu chung: Mỗi bài tập trong phần này sẽ được thực hiện lần lượt bằng cả 2 cách: Comprehension và Anonymous Function

129/. Viết chương trình cho nhập số nguyên n. Chương trình tạo ra 1 list L gồm n số nguyên âm hoặc dương ngẫu nhiên từ -100 đến +100. Tạo ra list M chỉ chứa những số trong L nhưng có giá trị >0 .

```
1 # c129trg52
2 import random
3 n = int(input("Vui long nhap so n : "))
4 lst = [random.randint(-100,100) for n in range(n)]
5 print((lst))
6 lstM1 = list(filter(lambda x: x>0 ,lst))
7 print((lstM1))
8 lstM2 = [x for x in lst if x>0]
9 print(lstM2)
```

```
1 import random
2 n = int(input('Nhập số nguyên n:'))
3 lstL = [random.randint(-100,101) for i in range(0,n)]
4 print(lstL)
5
6 lstD1 = list(filter(lambda x: x>0,lstL))
7 print(lstD1)
8
9 lstD2 = [x for x in lstL if x > 0]
10 print(lstD2)
11
12 lstA1 = list(filter(lambda x: x<0,lstL))
13 print(lstA1)
14
15 lstA2 = [x for x in lstL if x < 0]
16 print(lstA2)
```



```
1 import random
2 n = int(input('Nhập số nguyên n:'))
3 lstL = [random.randint(-100,101) for i in range(0,n)]
4 print(lstL)
5
6 lstD1 = list(filter(lambda x: x>0,lstL))
7 print(lstD1)
8 lstD2 = [x for x in lstL if x > 0]
9 print(lstD2)
10
11 lstA1 = list(filter(lambda x: x<0,lstL))
12 print(lstA1)
13 lstA2 = [x for x in lstL if x < 0]
14 print(lstA2)
15
16 print(lstD1+lstA1)
```

```
1 import random
2 n = int(input('Nhập số nguyên n: '))
3 lstL = [random.randint(-100,101) for i in range(0,n)]
4 print(lstL)
5
6 lstD1 = list(filter(lambda x: x>0,lstL)) + list(filter(lambda x: x<0,lstL))
7 print(lstD1)
8
9 lstD2 = [x for x in lstL if x > 0] + [x for x in lstL if x < 0]
10 print(lstD2)
11
12
```

132/. Tạo 2 listA, listB với kích thước khác nhau và chứa giá trị tùy ý. Ví dụ:

```
listA = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
```

```
listB = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]
```

- Viết chương trình trả về *listAinB* cho biết những giá trị nào trong *listA* có trong *listB* và ngược lại tạo 1 *listBinA* chứa những giá trị trong listB có trong listA.
- (*)Mở rộng: Sao cho mỗi list kết quả (*listAinB* và *listBinA*) không chứa giá trị trùng nhau. Thực hiện bằng 2 cách: cách 1 dùng set; cách 2 dùng dictionary.

```
1 lstA = [1,1,2,3,5,8,13,21,34,55,89]
2 lstB = [1,2,3,4,5,6,7,8,9,10,11,12,13]
3
4 lstAinB1 = list(filter(lambda x: x in lstB,lstA))
5 print(lstAinB1)
6
7 lstAinB2 = [x for x in lstA if x in lstB]
8 print(lstAinB2)
9
10 lstBinA1 = list(filter(lambda x: x in lstA,lstB))
11 print(lstBinA1)
12
13 lstBinA2 = [x for x in lstB if x in lstA]
14 print(lstBinA2)
```

temp.py

Bai101_Tr49.py Bai102_Tr49.py Bai129_Tr52.py Bai130_Tr52.py Bai132_Tr52.py temp.py

Buoi08 D:\TTTT_KHTN\Python

Bai18_Tr22.py Bai37_Tr34.py Bai38_34.py Bai86_Tr46.py Bai87_Tr46.py Bai88_Tr46.py Bai91_Tr47.py Bai97_Tr48.py Bai98_Tr48.py Bai101_Tr49.py Bai102_Tr49.py Bai129_Tr52.py Bai130_Tr52.py Bai132_Tr52.py temp.py

External Libraries

Scratches and Consoles

```
1 lstA = [1,2,3]
2 lstB = [3,3, 4,5]
3 lstAinB1 = list(filter(lambda x: x in lstB,lstA))
4 print(lstAinB1)
5 lstAinB2 = [x for x in lstA if x in lstB]
6 print(lstAinB2)
7 lstBinA1 = list(filter(lambda x: x in lstA,lstB))
8 print(lstBinA1)
9 lstBinA2 = [x for x in lstB if x in lstA]
10 print(lstBinA2)
```

97_Tr48 Bai98_Tr48 temp Bai102_Tr49

[3]

[3, 3]

[3, 3]

In[3]: